

GREP 빠른 참조

패턴 매칭, 정규식, 재귀 검색, 컨텍스트, 필터링

기본 사용법

grep 실행

```
grep "pattern" file.txt # search in file
grep "error" *.log # search multiple files
grep "hello" file1.txt file2.txt # explicit file list
cat file.txt | grep "pattern" # pipe input
dmesg | grep -i "usb" # filter command output
```

주요 플래그

- i 대소문자 무시 매칭
- v 반전 매칭 — 일치하지 않는 줄 출력
- c 일치하는 줄 수 출력
- n 줄 번호 표시
- l 일치하는 파일명만 나열
- L 일치하지 않는 파일명만 나열
- w 단어 전체 매칭만
- x 줄 전체 매칭만

정규식 패턴

기본 정규식 (BRE)

- .
- *
- ^
- \$
- [abc]
- [^abc]
- [a-z]
- <, \>
- \(\), \), \1

BRE 예시

```
grep '^#' file.conf # lines starting with #
grep 'errors' file.log # lines ending with error
grep '^$' file.txt # blank lines
grep 'co[ou]r' file.txt # match color or colour
```

확장 정규식

확장 정규식 (ERE)

- + 이전 요소 1회 이상
- ? 이전 요소 0회 또는 1회
- {n}
- {n,m}
- (a|b)
- ()

ERE 예시

```
grep -E '[0-9]{3}-[0-9]{4}' f # phone number pattern
grep -E '(error|warn|fatal)' f # multiple patterns
grep -E '(a-z){1,3}' f # capitalized words
grep -P 'd{1,3}\.d{1,3}' f # Perl regex: IP fragments
```

컨텍스트 줄

컨텍스트 예시

```
grep -B 3 "error" app.log # 3 lines before match
grep -A 5 "FAIL" test.log # 5 lines after match
grep -C 2 "crash" kern.log # 2 lines before and after
grep --group-separator="..." -C 1 "err" f # custom separator
```

컨텍스트 플래그

- B N
- A N
- C N
- group-separator=str
- color=auto

재귀 검색

재귀 예시

```
grep -r "TODO" # recursive from current dir
grep -rn "FIXME" src/ # recursive with line numbers
grep -r --include="*.py" "import" # only .py files
grep -r --exclude="*.log" "error" # skip .log files
grep -r --exclude-dir=node_modules "require" #
```

재귀 플래그

- r / --recursive
- R
- include=glob
- exclude=glob
- exclude-dir=dir
- include-dir=dir

카운팅 & 목록

카운트 & 목록 예시

```
grep -c "error" *.log # count matches per file
grep -l "TODO" src/*.py # list files with TODOs
grep -L "test" src/*.py # files missing "test"
grep -o "http[s]*" page.html # extract matching parts only
grep -c 'file.txt' # count total lines (like wc -l)
```

출력 플래그

- c
- l
- L
- o
- H / -h
- Z

반전 매칭

반전 & 제외

```
grep -v "^#" config.conf # remove comment lines
grep -v "^$" file.txt # remove blank lines
grep -v -e "debug" -e "trace" app.log # exclude two patterns
grep -v "pattern" f | grep "other" # chain: NOT A, then B
```

필터링 전략

- v
- v with -e
- pipe chain
- grep -v '^\$' | grep -v '^\$'
- v with -c

복수 패턴

복수 패턴 예시

```
grep -e "error" -e "warning" app.log
grep -E "error|warning|fatal" app.log
grep -f patterns.txt file.txt # patterns from file
grep -v -e "GET" -e "POST" access.log
```

패턴 옵션

- e pattern
- f file
- E 'a|b|c'
- F
- G
- P

성능

성능 팁

- F (fgrep)
- LC_ALL=C grep
- include/--exclude
- m N
- q
- ripgrep (rg)

성능 예시

```
LC_ALL=C grep -F "exact string" huge.log
grep -r -m 1 "needle" /var/log/ # stop after first hit
grep -rq "pattern" ; && echo "found" # boolean test
grep -r --include="*.go" "func main" #
```

일단 패턴

한 줄 명령

```
grep -rn "TODO\|FIXME\|HACK" src/ # find code markers
grep -oP '(?<=){1}(?=)' f # extract quoted strings
grep -E '\s$' f | wc -l # count blank lines
grep -c '.*.py' | sort -t: -k2 -rn # sort files by line count
grep -rn --include="*.yaml" "password" # audit for secrets
```

레시피

- IP addresses
- Email addresses
- URLs
- Lines between markers
- Unique matches
- Count per pattern